



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 06019861 A

(43) Date of publication of application: 28.01.1994

(51) Int. Cl. G06F 15/16

(21) Application number: 05087579
 (22) Date of filing: 14.04.1993
 (30) Priority: 30.04.1992 US 92 876670

(71) Applicant: INTERNATL BUSINESS MACH
 CORP <IBM>
 (72) Inventor: FERGUSON DONALD F
 GEORGIADIS LEONIDAS
 NIKOLAOU CHRISTOS N

(54) MECHANISM FOR DESIGNATION PATH OF TRANSACTION PROCESSING SERVER

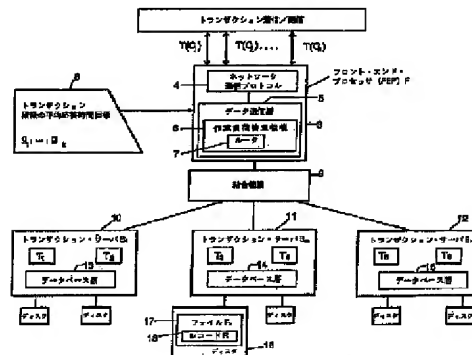
(57) Abstract:

PURPOSE: To adjust the load on each transaction processing server to attain the best responding performance in relation with an actual processing time across all hierarchies by classifying a transaction into plural hierarchies based on the request processing time, and setting a target responding time for every hierarchy in a multiprocessor transaction processing system.

CONSTITUTION: A load adjusting mechanism calculates a mean responding time for every present hierarchy, and the performance index of each hierarchy is calculated by considering a relation with the set target responding time of the hierarchy. When a new transaction arrives, the load adjusting mechanism inspects several transaction servers being several candidates whose paths can be designated by the trans-

action, and the performance index of every hierarchy when each is turned into the server is predicted. Then, an overall target established index obtained by synthesizing the performance indexes of all the hierarchies is derived, and the property of the transaction processing server whose path should be designated is evaluated.

COPYRIGHT: (C)1994,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-19861

(43) 公開日 平成6年(1994) 1月28日

(51) Int.Cl.⁵

G 0 6 F 15/16

識別記号

3 8 0 Z

庁内整理番号

8840-5L

F I

技術表示箇所

審査請求 有 請求項の数 4 (全 16 頁)

(21) 出願番号 特願平5-87579

(22) 出願日 平成5年(1993) 4月14日

(31) 優先権主張番号 8 7 6 6 7 0

(32) 優先日 1992年4月30日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72) 発明者 ドナルド・フランシス・ファーガソン

アメリカ合衆国11364、ニューヨーク州ベイサイド、トゥハンドレッド・ナインティーン・ストリート 61-29

(74) 代理人 弁理士 頓宮 孝一 (外4名)

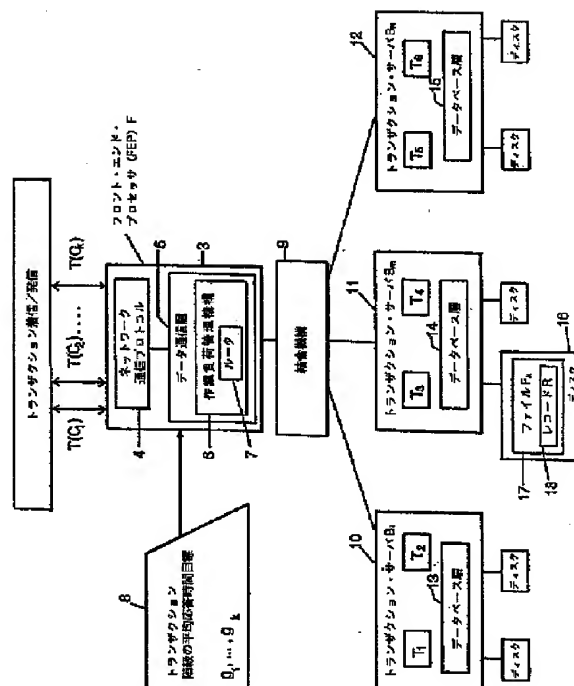
最終頁に続く

(54) 【発明の名称】 トランザクション処理サーバの経路指定機構

(57) 【要約】

【目的】 マルチプロセッサ・トランザクション処理システムにおいてトランザクションをその要求処理時間にもとづいて複数の階級に分類し、前記階級ごとに目標応答時間を設定し、階級全体について実際の処理時間との関係で最も応答性能がよくなるように各トランザクション処理サーバの負荷を調整する。

【構成】 負荷調整機構は現在の階級ごとの平均応答時間を算出し、その階級の設定目標応答時間との関係を考慮しつつ階級ごとの性能指標を算出する。新たなトランザクションが到着した時は、負荷調整機構はそのトランザクションが経路指定されることができいくつかの候補となるトランザクション・サーバを検討し、各々がサーバとなった時の階級ごとの性能指標を予測する。次に全ての階級の性能指標を統合した全体目標達成指標を導出した上で経路指定すべきトランザクション処理サーバの適否を評価する。



【特許請求の範囲】

【請求項1】複数の階級に区別されたトランザクションを単位として処理を行う複数のトランザクション処理サーバを有しているコンピュータ・システムにおいて、最適な応答時間を確保するために前記トランザクションを前記トランザクション処理サーバに経路指定する機構であって、

前記各々の階級ごとに現在の平均応答時間を計算する手段と、

前記各々の階級ごとに目標応答時間を記憶する手段と、 10

前記現在の平均応答時間と前記目標応答時間とにもとづいて、前記各々の階級ごとの現在の性能指標を求める手段と、

1の着信トランザクションにตอบสนองし、前記着信トランザクションを処理するために経路指定されうる全ての前記トランザクション処理サーバごとに、前記着信トランザクションが前記指定されうる前記トランザクション処理サーバのうちの1つにおいて処理される場合の、前記各々の階級ごとの前記性能指標を予測して予測性能指標を求める手段と、 20

前記予測性能指標に対応して、それらを全階級について統合した全体目標達成指標を前記経路指定されうる全てのトランザクション処理サーバごとに予測する手段と、

前記全体目標達成指標のうち最良のものを決定する手段と、

前記決定に対応する前記トランザクション処理サーバに前記着信トランザクションを経路指定する手段と、を含む経路指定機構。

【請求項2】1の階級に係わる前記階級ごとの現在の平均応答時間は、前記1の階級において処理が完了したトランザクションの到着時間と処理終了時間の差分値の時間重み付けされた平均値であることを特徴とした請求項1の経路指定機構。 30

【請求項3】前記予測性能指標を求める手段は、前記着信トランザクションの応答時間を予測する応答時間予測手段を有することを特徴とした請求項1の経路指定機構。

【請求項4】前記階級について前記現在の性能指標にもとづいて優先順位を付与し、その順に並べる手段と、前記各々のトランザクション処理サーバに経路指定された前記トランザクションを前記優先順位の順に処理する手段と、 40

をさらに含む請求項1の経路指定機構。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、一般にトランザクションを処理する相互接続されたコンピュータ・システムにおける作業負荷管理プログラムに関し、特に各々のトランザクション階級が個々の応答時間目標を持つ、異なる階級のトランザクションを有するコンピュータ・システム 50

ムにおける作業負荷管理プログラムに関する。

【0002】

【従来の技術】トランザクション処理システムは、一般に広範囲の地理的区域に分散させられた複数のターミナルをサポートする多重プロセッサ・データ処理システムで実行する、オン・ラインのアプリケーション指向のシステムである。典型的なトランザクション処理システムは、IBMのACP（エアライン制御プログラム）であり、航空会社予約システムで従来から使用されているが、また他のシステムでも使用され、特に銀行によってオン・ラインのテラー・アプリケーションにおいて使用されている。

【0003】従来の多量の作業は多重プロセッサ及び分散処理システムにおいて、作業負荷管理プログラムによって実行された。静的及び動的なロード・バランス・アルゴリズムに関しては、著者D. L. Eager, et al. による題名"Adaptive LoadSharing in Homogeneous Distributed Systems" (IEEE Transactions onSoftware Engineering, SE-12 (5), May 1986)、著者Y. T. Wang, et al. による題名"Load Sharing in Distributed Systems" (IEEE Transactions onComputers, C-34 (3), March 1985)、及び著者A. N. Tantawi, et al. による題名"Optimal Static Load Balancing in Distributed Computer Systems, (Journal of the ACM, pages 445-465, April 1985) で述べられているので参照されたい。しかしながら、上記のアルゴリズムは着信する全てのトランザクションの平均応答時間を最小にするのにとどまり、各々が異なる応答時間目標を有するトランザクションの異なる階級を考慮していない。

【0004】著者P. P. Bhattacharya, et al. による題名"Optimality and Finite TimeBehavior of an Adaptive Multi-objective Scheduling Algorithm" (Proc. ofthe 29th Conf. on Decision and Control, Honolulu, Hawaii, Dec. 1990) においては、ジョブ階級は割当てられた応答時間目標に該当し、スケジューリング・アルゴリズムはジョブをスケジュールするために提案されているが、これら全ては単一のプロセッサ内で実行される。上記記載のアルゴリズムは多重プロセッサ・システムへの応用においてどのように拡張されるのか或いは修正されるのか、及び複数のコンピュータに対して異なる応答時間階級目標を有するトランザクションの経路指定の追加された問題をどのように処理するかが示されていない。

【0005】静的及び動的なトランザクションの経路指定アルゴリズムの研究が行われ、その結果が、著者P. S. Yu, et al. による題名"Dynamic Transaction Routing inDistributed Database Systems" (IEEE Transactions on SoftwareEngineering, SE-14 (9), September 1988) 及び著者D. Cornell, et al. による題名"On Coupling Partitioned Database Systems" (Proceedings o

f 6th International Conference on Distributed Computing Systems, 1986) で報告されているので参照されたい。しかしながら、上記におけるトランザクション経路指定の研究は、また全着信トランザクションにおける平均応答時間を最小にするのにとどまり、階級化された応答時間を前提としていない。一般に全体的な応答時間の最小化は、階級の特定の目標を満足させない。

【0006】トランザクションの異なる階級のためには、異なる応答時間目標に従って作業負荷のバランスをとる、トランザクション多重処理システムの作業負荷管理プログラムを有することが望ましい。

【0007】

【発明が解決しようとする課題】本発明の目的は、応答時間が目標に達成しない程度を最小にするための作業負荷のバランスをとる相互接続されたトランザクション処理コンピュータ・システムのための作業負荷管理プログラムを提供することにある。

【0008】本発明の目的は、トランザクションが階級に分類され、各々の階級が個々の応答時間目標を持つシステムのための作業負荷管理プログラムを提供することにある。

【0009】本発明の目的は、個々の応答時間目標に対して性能不十分であったトランザクションの階級における応答時間未達成性能指標を最小にするように、上記システムのプロセッサ間において、作業負荷のバランスがとれるような作業負荷管理プログラムを提供することにある。

【0010】本発明の目的は、トランザクションの異なる階級の異なる応答時間目標を考慮しつつ、トランザクションの経路指定管理によって作業負荷のバランスをとることにある。

【0011】本発明の目的は、複数の経路指定をとった場合の各着信トランザクションの応答時間をそれぞれ予測することにある。

【0012】本発明の目的は、トランザクションのスケジューリングについて各階級毎に優先順位という概念を導入し、応答時間未達成性能指標に従ってこれらのスケジューリング優先順位を動的に調整することによって、作業負荷のバランスをとることにある。

【0013】本発明の目的は、スケジューリング優先順位を動的に調整することに伴う障害を考慮するために、トランザクションが経路指定された目的のプロセッサにおけるいかなるマルチプログラミングをも適用可能な経路指定管理によって作業負荷のバランスをとることにある。

【0014】本発明の目的は、特定の階級のトランザクションが必要とする記録に対して、より遅い平均アクセス時間をもつ他のプロセッサよりも、より速い平均アクセス時間を有するプロセッサによって、特定の階級のトランザクションがより能率的に実行できることを事前に

考慮して作業負荷のバランスをとることにある。

【0015】

【課題を解決するための手段】上記目的及び利点は、トランザクションを処理する複数のサーバと着信トランザクションをサーバに分散させるための作業負荷管理プログラムを具備する本発明によって得られる。トランザクションをその応答時間目標によって、いくつかの階級に分類する。

【0016】トランザクション階級はトランザクション・プログラム名、トランザクション・コード、ユーザ、トランザクションを提示するターミナル、またはワークステーション、及び多数の他の要素にもとづいて分類されたトランザクションのグループである。階級応答時間目標はオペレータによって外部から指定する等の方法で設定される。

【0017】作業負荷管理プログラムは各トランザクション階級の現在の階級平均応答時間 (average class response time) を計算し、各階級目標応答時間との関連において、各階級の現在の階級性能指標を導き出す。好ましい実施例では、ある階級の現階級性能指標は(階級平均応答時間)(階級応答時間目標)によって表される。トランザクション階級の現平均階級応答時間及び各階級の現階級性能指標は、トランザクションが処理完了するたびに更新される。

【0018】トランザクションが着信する度に作業負荷管理プログラムは、着信トランザクションがとりうる多数の経路を熟慮し、可能な経路のそれぞれについて階級性能指標を予測する。好ましい実施例では、全ての可能な経路が熟慮される。全資源消費量、全I/O遅延時間、及び全通信遅延時間がそれぞれ経路指定に対して予測される。全体目標達成指標が各々の経路に対して求められ、最良の全体目標達成指標が得られる経路の選択が行われる。好ましい実施例では、可能な経路指定選択の全体的な目標達成指標は該経路指定選択の予測された最悪(すなわち最高位)の階級性能指標であり、最良の全体的な目標達成指標は、これらの最悪の階級性能指標の最良(すなわち、最低位)の1つである。これは最小/最大機能と呼ばれることがある。

【0019】好ましい実施例では、作業負荷管理プログラムは、また性能の良くない階級のトランザクションが、経路指定された宛先のサーバにおいて、性能の良い階級のトランザクションよりもさらに高位のタスク指名順位が得られるように、現階級性能指標、現平均階級応答時間と階級応答時間目標との比率に従って階級を優先設定する。

【0020】各々のサーバは互いに機能的に独立しており、別々のコンピュータで実行される。所望するならば、物理的に同じコンピュータによって複数のサーバを実行できる。ルータ機能は、また各サーバから機能的に独立しており、別々のコンピュータまたは1つ以上のサ

5

ーバを実行するコンピュータによっても実行できる。

【0021】

【実施例】

【数1】

$$\vec{N}$$

は以降ベクトルNと記載する。本発明は、多重プロセッサ・トランザクション処理システムにおいて、高性能の動的な作業負荷管理を実行するために新しい技術を実施する。各トランザクションは一般にユーザ入力データを処理し、1つ以上のデータベースに対して読出し及び更新を実行し、アプリケーション・コードを実行してその結果をユーザに送信する。ユーザと実行中のトランザクションとの間で複数の対話ができる。

【0022】システム・ハードウェア・アーキテクチャの好ましい実施例が、図1に示されている。システムで実行される作業は、データベース要求を実行するユーザ提示のトランザクションである。著者C. J. Date、による題名"An Introduction to Database Systems" (Volume II, Addison Wesley Company, Reading Mass., 1983) は、典型的なトランザクション及びデータベース要求実行を説明しているので参照されたい。各々の提示されたトランザクションは予め定義されたトランザクション階級に属する。図1は6つの異なるトランザクション T_1 、 T_2 、 T_3 、 T_4 、 T_5 、 T_6 を示し、トランザクション・サーバ B_1 、トランザクション・サーバ B_n 及びトランザクション・サーバ B_x によって実行される。 C_1 、 C_2 、 \dots 、 C_k で表されるK個のトランザクション階級がある。階級 C_1 の着信または発信のトランザクションは、例えば $T(C_1)$ として表される。ある階級での着信トランザクションの識別は、トランザクション・プログラム名、トランザクション・コード、ユーザ及び多数の他の要素に依存する。各々のトランザクションはユーザ入力データを処理し、データベースに対して読出し及び更新を実行し、アプリケーション・コードを実行してその結果をユーザに送信する。ユーザと実行中のトランザクションの間では複数の対話ができる。

【0023】トランザクション処理システムはフロント・エンド・プロセッサ(FEP)F及びN個のトランザクション・サーバ B_1 、 B_2 、 B_n 、 \dots 、 B_x を有する。各々のサーバは完全に機能的なコンピュータ・システムであり、CPU、メモリ、ディスク、その他を有する。これらのコンピュータは結合機構9を介して、例えば、高帯域幅相互接続ネットワーク等に相互接続される。Fで表されるプロセッサはフロント・エンド・プロセッサ(FEP)として指定され、ルータ7を含む作業負荷管理プログラム6を有する。FEPはユーザとの通信に必要なネットワーク通信プロトコル4を実行する。FEPは、またデータベース・システムの機能であるデータ通信層5を実行する。著者C. J. Date、による題

6

名"An Introduction to Database Systems" (Volume I, Addison Wesley Company, Reading Mass., 1983) は、データ通信層がソフトウェアとどのようにインタフェースするかを述べている。このソフトウェアはデータベースのユーザによって使用されるターミナル、またはインテリジェント・ワークステーションとのメッセージのやりとりをするために送受信するプロセッサの通信ハードウェアを制御する。ユーザによって発信されたメッセージは、データ通信層によって特定の種類のトランザクションを開始させる。実行中のトランザクションから発信されるユーザへの応答がメッセージに組込まれてユーザに送信される。データ通信層5の構成要素の1つは作業負荷管理プログラム6である。作業負荷管理プログラム6の機能の1つは、例えば T_1 、 \dots 、 T_6 を始めとするトランザクションを適切なトランザクション・サーバに経路指定することにある。これは、作業負荷管理プログラムの構成要素であるルータ7によって実行される。全ての着信及び処理完了のトランザクションはFEPを通る。本明細で述べる技術は、一般に複数のFEPの場合である。

【0024】その他のプロセッサは、バック・エンド・プロセッサ(BEP)としての役目をし、トランザクション・サーバと呼ばれる。作業負荷管理プログラム及びトランザクション・サーバが、同一プロセッサ(コンピュータ)に存在するシステム構成にすることは非常に良いことである。これらのプロセッサはトランザクション・プログラムを実行し、プログラムによって行われるDB要求を処理する。これらの機能は同時実行制御、指標付け、バッファリング、ディスク入出力、その他を有し、及びデータベース層13乃至15を有する。本発明の実施例によって仮定されるモデルにおいて、データベースの記録はBEP全体に区分される。プロセッサ B_1 で実行しているトランザクション T_1 が、ファイル(関連DB) $F_k(17)$ のレコード $R(18)$ に関して何らかの処理を要求したものと仮定する。そのレコードが B_1 内に存在していれば、その動作は局部的に実行される。さもなければ、動作はレコードを有するプロセッサ B_n に機能移管される。著者D. W. Cornell, et al. による題名"On Multisystem Coupling through Function Request Shipping" (IEEE Transactions on Software Engineering, SE-12 (10), October 1986.) は、上記方法がIBMトランザクション処理用の製品であるCICSによってどのように行われるかを述べる。 B_n はその動作を処理し、結果とデータを B_1 に送信し、 B_1 はその結果を T_1 に送る。ここでの実施例では、BEP全体でデータベースを区分していると仮定しているが、経路指定アルゴリズム(及び、本実施例の一部であるスケジューリング・アルゴリズム)は、またデータベースがBEP間で共有される、共有データベース環境にも適用できる。

【0025】好ましい実施例によれば、データベース管理者は各トランザクション階級において平均応答時間目標を定義する。これらの目標は、 g_1, g_2, \dots, g_i で表される。問題はこれらの目標を達成するためにトランザクション作業負担を管理することである。トランザクション階級の性能目標を定義して実行させる能力は、多様なユーザ・コミュニティによって提示された様々な種類のトランザクションを処理するシステムにとって重要である。さらに処理レベル協定が作業の異なる階級のために必要な応答時間の観点から通常、指定される。著者C. Watson, et al. による題名“The Three Phases of Service Levels” (MainframeJournal, July 1989.) は、この業界で使用される様々な処理レベル協定の例を与えるので参照されたい。

【0026】本発明の実施例では2種類の作業負荷管理方法を用いる。第1はFEPで実行されるトランザクション経路指定アルゴリズムの使用である。各々の着信トランザクションはトランザクションによって、要求された推定資源にもとづくBEPの1つに経路指定され、BEPの現システム状態（ロード）及び応答時間目標への達成度は、その経路選択の結果として満足させられる。第2は動的優先順位方式の使用である。この方式はBEPのスケジューリング優先順位を動的に調整する。これらの優先順位はCPU処理のために待機中のどのトランザクションをタスク指名するか及び割込みできるか、できないかを決める。

【0027】経路指定技術について説明する。トランザクション経路指定技術はFEPのデータ通信層で実行される。この経路指定技術は、経路指定を求めるのに必要な3つの集合の制御変数を管理する。これらは次の通りである。

【0028】1. ベクトル $N = \{N_1, N_2, \dots, N_i\}$: フィールド N_i は現在、システムにある（BEPへ経路指定された）階級 C_i のトランザクション数であ*

$$P(j, l) = \alpha \cdot P_j + (1 - \alpha) \cdot N_j \cdot \left(\frac{R(S, i, l, j)}{g_j} \right).$$

【0036】この計算は図5に示されている。値 $P(j, l)$ は、 T がBEP B_l に経路指定された場合における、階級 C_j の性能指標の予測された新しい値である。 T が B_l に経路指定される場合、現在 B_l に対してアクティブ（経路指定されている）である階級 C_j のトランザクションの応答時間に影響を及ぼす。 T は、また他のBEPに対して機能移管を生じさせ、他のBEPに経路指定されている階級 C_j のトランザクションの応答時間に影響する。パラメータ α は $0 \leq \alpha \leq 1$ の範囲であり、現システム状態と以前に観測された応答時間（過去の経歴）との比較の重要性に重みをおくのに利用される。

【0037】3. トランザクション T は次式を満足する

*る。

【0029】2. ベクトル $P = \{P_1, P_2, \dots, P_i\}$: フィールド P_i は階級 C_i の性能指標と呼ばれ、階級 C_i の現目標達成度を表す。 P_i は現平均階級 C_i の応答時間を目標 g_i によって除算した値である。 $P_i \leq 1$ ならば、階級 C_i はその目標にかなう。性能指標が小さいほど性能が良好で目標が十分に達成されていることを意味する。ベクトル P を管理する技術は後で説明する。

【0030】3. S はシステム状態情報を表す。この情報は過去の経路指定決定の経歴、BEPの現ロード、トランザクション階級の資源使用統計値、その他を有する。この状態情報は経路指定技術の異なる実施によって変わる。性能予測副構成のために必要である状態情報の実施例が後で述べられる。

【0031】経路指定技術は副機能 $R(S, i, l, j)$ を用いる。 S は現システム状態である。 i は最後に着信したトランザクションの階級ID（識別子）である。 l はBEP ID、及び j は任意の階級IDである。この副機能は着信トランザクション $T \in C_i$ がBEP B_l に経路指定される場合、既にシステムに存在する階級 C_j のトランザクションの平均応答時間を予測する。 R の実施例は後で述べる。

【0032】経路指定技術は各トランザクションがFEPに、着信及び各トランザクションの処理完了毎に呼出される。

【0033】着信アルゴリズムについて説明する。着信トランザクション $T \in C_i$ があると仮定する。以下のステップが実行される。

【0034】1. フィールド $T_arrival_time$ の着信時間 T を記録する。

【0035】2. 全ての階級 C_i 、及び全てのBEP B_l を計算する。

【数2】

BEP B_l に経路指定される。

【数3】

$$\min_l \max_j \{P(j, l)\}$$

（ l を固定して $P(j, l)$ をすべての j について計算し、そのときの最大値を $Q(l)$ とすると、 $Q(l)$ を最小にするものをいう。）

【0038】この手法は全ての性能指標を等しくさせ、且つ可能な限り小さくさせる。この手法によってある階級が目標を大きく上回ったために、他の階級が目標を達成できない、という事態の到来を防止する。

【0039】4. T が B_l に送信された後、経路指定を反映するために N_i 及び S は更新させられる。

9

【0040】発信アルゴリズムについて説明する。トランザクション T C_i の処理完了(発信)があると仮定する。図6は発信アルゴリズムの図式である。以下のステップが実行される。

【0041】1. $R = \text{Current_time} - T.$
 arrival_time を設定する。

【0042】2. 次式により P_i の値を更新する

【数4】

$$P_i = \beta \cdot P_i + (1 - \beta) \cdot \left(\frac{R}{g_i} \right).$$

【0043】パラメータ β は過去の応答時間と最後に発信した応答時間との相対的な重み付けを示す。その他の技術では最後に発信した階級での平均応答時間を計ることにより、 P_i を計算することが可能である。

【0044】3. N_i 及び S を更新する。

【0045】動的優先順位付け技術について説明する。FEP/DC層装置は正確に目標達成度をモニタできる、トランザクション処理システムの唯一の構成要素である。従って、FEP/DC層装置はBEPで使用されるトランザクション階級の優先順位を設定できる。BEPは局所的な情報にもとづく優先順位を設定することはできない。これは、各々のBEPはトランザクション階級内の一部分のみの処理完了をモニタするにすぎないからである。従って、目標達成度に関するBEP情報は非常に不正確である。

【0046】動的優先順位付け技術は3つの構成機能から成る。第1の機能は目標達成度にもとづく階級優先順位を更新し、その更新をBEPに知らせるFEP/DC機能である。第2の機能はFEP/DCによって送信された優先順位の更新を受信して処理するBEPの副機能である。第3の機能はBEP内でスケジューリング手法を実行する副機能である。この第3の副機能はどのトランザクションがCPUを占有するかを決める。これらの3つの機能について次に述べる。

【0047】FEP/DC更新アルゴリズムについて説明する。FEP/DC更新アルゴリズムはトランザクションが処理完了する度に呼出される。これは前述の発信アルゴリズムのステップ2に続く。このアルゴリズムは内部制御変数 $o(i)$ 、 $i = 1, 2, \dots, K$ を管理する。集合 $o(i)$ は集合 $\{1, 2, \dots, K\}$ の置換であり、最大性能指標から最小性能指標にソートされたトランザクション階級IDを表す。言い換えると、集合 $o(i)$ は次式の特性を有する。

【数5】 $P_{o(i)} \geq P_{o(i+1)}$

【0048】処理完了のトランザクション $T \in C_i$ があると仮定する。優先順位更新アルゴリズムは性能指標 P_i の更新後、以下のステップを実行する。

【0049】1. $P_{o(i)} \leq P_{o(i+1)}$ または $P_{o(i)} \geq P_{o(i-1)}$ ならば、

10

a. 性能指標の現在値にもとづいて順序、 $o(1)$ 、 $o(2)$ 、 \dots 、 $o(K)$ を再計算する。

b. ベクトル $P = \{P_1, P_2, \dots, P_K\}$ の新しい値を全てのBEPに送信する。

【0050】順序リスト $o(i)$ はオーバーヘッドを、より低くするのに使用される。B-treeなど、性能指標が高い階級を探索するオーバーヘッドをより低くする他のいくつかの実施方法がある。階級の相対的順位を実際に変える性能指標の変化だけがBEPに送信される。順位付けを変えない変化は、BEPの優先順位に影響しないので送信されない。さらにオーバーヘッドをより低くするには、FEP/DCによって経路指定されたトランザクションの更新された性能指標をピギーバック(piggyback=トランザクションのメッセージの特別なフィールドに挿入)することで可能である。これによってトランザクションがBEPに送られたとき、それとともに新しい性能指標も送ることができるので余分なメッセージが不要となる。本明細書の実施例では、FEP/DCがトランザクションをBEPに経路指定する度に、BEPは最新のベクトル P が送信されたかをテストする。テスト結果が否定ならば、新しいベクトル P が経路指定されたトランザクションに付加され、BEPがトランザクションを受信するときにBEPによって抽出される。この最適化が優先順位を更新するための余分なメッセージの送信の必要性を排除する。オーバーヘッドは更新された性能指標ベクトルが送信される間において、処理完了の数を固定することを要求することによってさらに減らすことができる。

【0051】性能指標ベクトルをBEPに送信する他の代替方法は、新しく経路指定されたトランザクションの階級の優先順位のみ更新することである。例えば、階級 C_1 のトランザクション T_1 が着信すると仮定する。FEPは T_1 をBEP B_3 に経路指定し、BEP B_3 の階級 C_1 の優先順位を P_1 から P'_1 に変更することを決めたと仮定する。次にメッセージ $\langle P'_1, T_1 \rangle$ はBEP B_3 に送られる。

【0052】受信更新アルゴリズムについて説明する。各々のBEPはベクトル P の局所のコピーを管理する。ベクトル P の新しい値を含むメッセージ(またはトランザクション)を受信後、局所コピーが更新される。

【0053】BEPスケジューリング・アルゴリズムについて説明する。BEPスケジューリング・アルゴリズムは実行中のトランザクション T がCPUを解放して待ち状態に入る度に呼出される。トランザクションはI/O待ち、機能移管応答、ユーザ入力、ロック、その他を含む多くの理由で待ち状態に入る。BEPスケジューラが割り込み可能で、より高位の優先順位である階級に属する他のトランザクションが実行可能になった場合、実行中のトランザクションは待ち状態におかれる。待ち状態が完了したのちに、 T は実行準備をさせられ、CPUに

対して待機する。BEPは全てのトランザクションのトランザクション階級を知っている。機能移管要求の階級は、その要求が実行されるトランザクションの階級と同じである。

【0054】BEPスケジューリング・アルゴリズムは少なくとも1つの待機トランザクションを有している階級中で最も高位の優先順位を有する階級 C_i からトランザクションをタスク指名する。これはラウンド・ロビン・スキャンを使用してトランザクション階級を調べることによって実行される。ラウンド・ロビン・スキャンは同一優先順位を有する他の階級と、階級 C_i とを区別するために使用される。階級の優先順位は性能指標によって決まる。一般に階級の性能指標が高くなると、その優先順位が高くなる。しかしながら、性能指標が非常に接近している階級に、厳密な優先順位を与えることを避けるために一定範囲の性能指標を有する階級の一群は同一の優先順位であると規定される。

【0055】実行中のトランザクションが待ち状態に入る度に、図4にも示されているように以下のステップが実行される。

【0056】a. $count = 0$

【0057】b. $i = last_dispatched$
{ $last_dispatched$ は、スケジューリング・アルゴリズムによってタスク指名された最後のトランザクションの階級IDを記録する。初期設定は1である。}

【0058】c. $best_class = -1$

{アルゴリズムは、待機トランザクションを有する最高位の優先順位の階級を探すために全てのトランザクション階級を調べる。}

【0059】d. $count \leq K$ の間

1) $i = i + 1$ を実行する。{階級はラウンド・ロビンによってスキャンされる。指標が一回りすると、指標は1に設定される。}

2) もし $i > K$ ならば

a) $i = 1$

{アルゴリズムが実行可能状態にあるトランザクション、または調べられた階級がより高位の優先順位を有している場合は、これまでに調べた最高位の優先順位の階級を現階級に設定する。}

3) CPU処理のために待機している階級 C_i のトランザクションがある場合、

a) もし $best_class = -1$ OR $P_{best_class} < P_i - \epsilon$ ならば { ϵ は群パラメータである。 $\epsilon = 0$ ならば、優先順位は性能指標によって厳密に求められる。さもないければ、優先順位は性能指標が ϵ の範囲内のトランザクションについては一定である。}

i. $best_class = i$

4) $count = count + 1$

{もうひとつの階級が調べられる}

EndWhile; {この時点で、タスク指名される次のトランザクションを有する階級が特定されている。この手法は、単にトランザクションを階級から拾い上げてタスク指名する。}

【0060】e. もし $best_class \neq -1$ ならば以下を実行する。

1) 待機トランザクションTを階級 C_{best_class} から拾い上げる

2) タスク指名 (T)

3) $last_dispatched = best_class$

【0061】このアルゴリズムは非割込みのスケジューリング手法を実行する。BEPスケジューリング・アルゴリズムに対する1つの可能な強化は、強制排除を実行することである。トランザクション $T \in C_i$ が実行準備されたと仮定する。全ての実行中のトランザクションは最下位の優先順位のトランザクション $T' \in C_j$ を見つけるために調べられる。 $P_i - \epsilon > P_j$ ならば T' は待機に戻され、Tがタスク指名される。

【0062】トランザクション処理システムのある実施形態において、BEPは内部マルチプログラミング・レベル制約を実施する。この実施形態において、最大で M_{ik} 個のトランザクションがどの時点でもBEP B_k でアクティブである。トランザクションはトランザクションがプロセスに連結され、CPUに対しての待機、DB要求等が許される限りアクティブであると定義する。非アクティブ・トランザクションはプロセスに連結されアクティブになる前に、アクティブなトランザクションが処理完了するのを単に待つ。マルチプログラミング制約は、通常、メモリ制限及びロック・コンテンションを避けるために考慮される。上記のスケジューリング・アルゴリズムは、またアクティブなトランザクションが処理完了した時に、待機している非アクティブなトランザクションのうちのどれが、アクティブにされるかを決めるために使用される。階級 i のトランザクションまたは機能移管要求が B_k に着信するときに、 B_k にすでに M_{ik} のアクティブなトランザクションがあれば待機させられる。

【0063】図2は任意のトランザクション・サーバBEP B_k 内の待ち行列の構造を示す。トランザクション・サーバは複数のCPU(2つのCPU 20、21が示されている)を有することができる。各トランザクション階級は2つの待ち行列を有する。第1の待ち行列はCPU処理のために待機しているアクティブなトランザクションを含む。第2の待ち行列は処理完了を待つ非アクティブなトランザクションを含む。例えば、階級 C_1 はCPU処理を待つ階級 C_1 の全てのアクティブなトランザクションを含む待ち行列22と、処理完了を待つ階級 C_1 からの非アクティブなトランザクションを有する待ち行列25とを有する。

【0064】 応答時間予測装置について説明する。応答時間予測装置は着信が生ずる毎に、経路指定技術によって呼出される副構成要素である。この構成要素はシステムに現在ある全トランザクションが一定の経路指定をされたときにその応答時間の変化を予測する。本明細では、この構成要素の1実施例がBEP間にて区分されたデータベースを仮定して提案される。共用データ・ベースの場合でも、応答時間予測装置を有することが可能である。他の多数の実施例が可能であり、関連方法が著者 C. T. Hsieh, et al. による題名“A Noniterative Approximate Solution Method for closed Multichain Queueing Networks”、(Performance Evaluation, 9: 119-133, 1989.) 及び著者 P. S. Yu, et al. による題名“Dynamic Transaction Routing in Distributed Database Systems” (IEEE Transactions on Software Engineering, SE-14 (9), September 1988.) に記載されているので参照されたい。本実施例は従来技術に対して、2つの主な新考案がある。第1に、従来技術はBEPにおけるトランザクション階級の優先順位を動的に変更処理しない。第2に、本実施例はほとんどの従来技術よりも低いオーバーヘッドを提供する。

【0065】 以下は予測装置への入力である。

- a. Sは現システム状態を示す。以下は本実施例によって用いられる状態情報を表す。
- b. iはトランザクション階級IDである。
- c. lはBEPのIDである。
- d. jはトランザクション階級のIDである。

予測装置は階級 C_i からの着信トランザクションがBEP B_l に経路指定される場合、システムに現在ある階級 C_i のトランザクションの平均応答時間を予測し、出力する。

【0066】 状態情報について説明する。本実施例の予測装置によって用いられる状態情報Sは、システムに既に存在するトランザクションについての情報を有し、階級及びBEPごとのトランザクション資源消費を予測する。Sの第1の構成要素は、システムに既にあるトランザクション数を表す。これは、 $M(i, l)$ で表され、システム中に存在するBEP B_l に経路指定されたトランザクションのうち階級 C_i のものの数を記録する。この行列は着信トランザクションがBEPに経路指定されるか、またはシステムのトランザクションが処理完了する毎にFEP/DCによって更新される。

【0067】 他の状態情報はトランザクション階級の資源消費についての統計情報である。この情報はBEPのルーチンをモニタすることによって集められる。各々のBEPは最後に観測された統計値を、複数のBEPによって報告された統計値をマージするFEP/DCへ定期

的に転送できる。通常、トランザクション処理システムのDBプロセッサは、トランザクション資源消費に関する統計値を集める。この情報はオフ・ライン性能調整、容量計画、及び原価計算に用いられる。

【0068】 FEP/DCはSの一部として以下の資源消費統計値を管理する。

【0069】 a. $W(i, l, k)$: これはBEP B_l に経路指定された階級 C_i のトランザクションによってBEP B_k で行われたCPU作業の平均値である。 B_l で実行されたCPU時間はDB要求の処理時間だけでなく、適用業務処理のCPU時間をも含む。CPU作業は、また機能移管及び遠隔のDB要求の処理による $k \neq l$ の場合の他のBEP B_k でも発生する。この行列の要素は、階級 C_i のトランザクションに起因する全てのCPU作業を含む。

【0070】 b. $V(i, l, k)$: この要素は階級 C_i のトランザクションがBEP B_l に経路指定された、すなわち、BEP B_k を“訪問”した平均回数を記録する。 $k \neq l$ ならば、その訪問は B_k に対する機能移管のDB要求(準備/実行処理を含む)である。 $l = k$ の場合、訪問はトランザクションによって生成されたDB要求の数に限定される。この合計は局所及び遠隔での実行と、局所及び遠隔のDBの読出し及び書込みを含む。

【0071】 c. $I(i)$: これは階級 C_i のトランザクションの予測全I/O遅延時間である。これはI/Oサブシステムの遅延時間だけを含む。データが区分化されているために、 $I(i)$ はトランザクションに対する経路指定の決定に依存しない。

【0072】 d. $D(i, l)$: これはBEP B_l に経路指定された階級 C_i のトランザクションの予測全通信遅延時間である。この遅延はBEPを接続しているネットワークを介しての機能移管DB要求、及び実行/準備メッセージをも含む。

【0073】 e. $C(i, l)$: これは B_l に経路指定された階級 C_i のトランザクションに対する全ての位置におけるログ(実行/準備)記録の書込みの予測全I/O遅延時間である。この遅延時間は標準の2段階の実行プロトコルにおける、様々な最適化のための経路指定の決定に依存する。

【0074】 推定アルゴリズムについて説明する。推定アルゴリズムは予測をするために状態情報を用いる。アルゴリズムはSの資源消費統計値が更新される毎に、以下の内部行列を予め計算する。

【0075】 a. $T(i, l)$: これは B_l に経路指定された階級 C_i のトランザクションによって消費された資源の和である。これは次式によって与えられる。

【数6】

$$T(i, l) = \left(\sum_{m=1}^N W(i, l, m) \right) + D(i, l) + I(i) + C(i, l).$$

【0076】これは待ち合わせ遅延時間がない（すなわち、システムは空である）場合の、 B_1 に経路指定された階級 C_1 のトランザクションの応答時間に近似する。

b. $B(i, l, k)$: これは B_1 に経路指定された階級 C_1 のトランザクションに係わる B_k での平均CPU”バースト長さ”である。これは、次式によって近似する。

【数7】

$$B(i, l, k) = \frac{W(i, l, k)}{V(i, l, k)}.$$

【0077】これらの2つの行列は半静的でシステムの異なる階級のトランザクション数に依存しない。

【0078】応答時間予測装置は図3で概略的に示されているが、これについて説明する。階級IDは $P_1 \geq P_2 \geq \dots \geq P_K$ の順序であると仮定する。これによって説明は簡単になるが、このことはアルゴリズムにとっての本質ではない。アルゴリズムは次に述べる通りである。

【0079】a. 内部変数を初期化する。

1) $Q(m) = 0$; $m = 1, 2, \dots, N$

{ $Q(m)$ はトランザクションが B_m を訪問する場合、CPUを得るまでトランザクションが待機する時間の量を表す。この量は正確に計算するのは難しく厄介であるので概算値を用いる。アルゴリズムは最高位の優先順位の階級から始まり、 $Q(m)$ をゼロに初期化する。アルゴリズムは階級 C_1 のトランザクションが階級 C_j ($j = 1, \dots, i$)のシステムに現存する全てのトランザクションが処理されるまで待機しなければならないという事実を考慮して、 $Q(m)$ を連続して更新する。}

2) $RT(j, k) = 0$; $j = 1, 2, \dots, K$; $k = 1, 2, \dots, N$

{ $RT(j, k)$ はプロセッサ B_k に経路指定された階級 j のトランザクションの平均応答時間である。}

3) $RP(j, k, m) = 0$; $j = 1, 2, \dots, K$; $k = 1, 2, \dots, N$; $m = 1, 2, \dots, N$

{ $RP(j, k, m)$ は、 B_k に経路指定された階級 j のトランザクションが BEP_m に存在する可能性の推定値である。}

a) $Q(m) =$

$$Q(m) + \left(\sum_{k=1}^N M(j, k) \cdot RP(j, k, m) \cdot B(j, k, m) \right).$$

【0080】b. 提示された経路指定を反映するために、 $M(i, l)$ の局所のコピーを一時的に増加させる。{様々なトランザクションに対する予測待ち合わせ遅延時間をここで計算する。優先順位の関係から階級 C_1 のトランザクションは、階級1、2、 \dots 、 $j-1$ からのトランザクションの後にのみ待機する。トランザクションの応答時間は、その処理要求と経験した待ち合わせ遅延時間とで求められる。}

【0081】c. FOR $j=1$ TO K DO

1) FOR $k=1$ TO N DO

{ BEP への訪問に起因する階級/経路指定ごとの待ち合わせ遅延時間分ごとにシステムの期待時間を増加する。}

【数8】

a) $S(j, k) =$

$$T(j, k) + \sum_{m=1}^N V(j, k, m) \cdot Q(m).$$

【0082】{階級 C_1 のトランザクションが BEP_{B_m} でCPU処理を待っている予測された確率を更新する。これは B_m における処理要求と待ち合わせ遅延時間に比例し、全システム時間に反比例すると考えられる。}

2) FOR $k=1$ TO N DO

a) FOR $m=1$ TO N DO

【数9】

i. $RP(j, k, m) =$

$$\frac{W(j, k, m) + V(j, k, m) \cdot Q(m)}{S(j, k)}$$

【0083】{この階級によって生じた追加の待ち合わせ時間の量だけ、予測待ち合わせ遅延時間を増加させる。これは平均処理バースト長さに対してCPU処理のために待ち合わせする確率を乗算して近似を得る。}

3) FOR $m=1$ TO N DO

【数10】

17

【0084】4) 処理要求及び待ち合わせにもとづく、 B_k に経路指定された C_i のトランザクションの応答時間を計算する。

FOR $k=1$ TO N DO

【数11】

a) $RT(j, k) =$

$$T(j, k) + \left(\sum_{m=1}^N Q(m) \cdot V(j, k, m) \right).$$

【0085】d. この時点で、現在システムにある全トランザクションの応答時間が予測された。予測装置によって復帰された平均値は、加重平均として与えられる。

FOR $j=1$ TO N DO

【数12】

$$R(S, i, l, j) = \frac{\sum_{k=1}^N RT(j, k) \cdot M(j, k)}{N_j}.$$

【0086】応答時間予測装置の計算コストをより低くできる複数の潜在的な最適化がある。第1は B_i への C_i の任意の経路指定において、アルゴリズムによって全ての j に対して、1回の通過で $R(S, i, l, j)$ を計算することが可能である。第2は階級 C_i に対しての経路指定の決定は、 $P_j > P_i$ での何れの階級 C_j における結果に対して影響を及ぼさない。この基本的な方式に対する多数の他の最適化が可能である。上記で提案されたのは2つの点において異なる、計算の少ないオーバーヘッドであるアルゴリズムを次に述べる。第1はアルゴリズムは $BEBB_k$ でCPUを待っている階級 C_i のトランザクションの確率を計算しない。その代わりに、アルゴリズムは階級 C_i の $Q(j, k)$ による $BEBB_k$ の作業量を計算し、トランザクションがCPUに連結される前のトランザクションが $BEBB_k$ で待たなければならない時間量が、 $Q(1, k) + \dots + Q(i, k)$ であると仮定する。第2はアルゴリズムは全ての階級の経路指定の効果を予測しない。その代わりに、トランザクションの推定応答時間が最小である BEP に対してトランザクションを経路指定する。

【0087】簡素化アルゴリズムについて説明する。前述のように、階級 C_i のトランザクションは FEP に着信し、トランザクションが B_i に経路指定された場合にトランザクションの応答時間に対する推定が所望されると仮定した。ここでもトランザクション指標は $j > i$ の場合にトランザクション j がトランザクション i より高位の優先順位を持つように、順序付けされると仮定する。

18

【0088】推定アルゴリズムについて説明する。

a. 内部変数を初期化する。

1) $Q(j, k) = 0$: $j=1, \dots, K, k=1, \dots, N$

b. FOR $j=1$ TO i DO

1) FOR $k=1$ TO N DO {階級 C_i による B_k の作業量を予測する。}

【数13】

$$Q(j, k) = \sum_{m=1}^N M(j, m) \cdot W(j, m, k)$$

【0089】c. {着信トランザクションが B_i に経路指定される場合、着信トランザクションの応答時間を計算する。}

【数14】

$$RT(i, l) = T(i, l) + \sum_{k=1}^N V(i, l, k) \left(\sum_{j=1}^i Q(j, k) \right)$$

【0090】経路指定アルゴリズムを説明する。

a. トランザクションは、 $RT(i, k)$ が最小となるように、 B_i に経路指定される。

b. $M(i, 1) = M(i, 1) + 1$ { B_i における C_i のトランザクション数を1つ増加させる。}

【0091】本発明は、何れの多重プロセッサ・トランザクション処理システムで実行できる。本明細に記述された本発明は、区分されたデータ（共有なし）のデータベース・システムに最も直接に適用できる。本発明は応答時間予測装置を修正することによって、データ共有アーキテクチャにも適用できる。トランザクション経路指定技術及び動的優先順位付け技術は、データ共有アーキテクチャに対して変わることはない。本発明は又、周期的に性能情報を共有するDCを有することによって、複数のFEP/DCのトランザクション処理システムにも応用できる。

【0092】本発明は、好ましい実施例に関して図示して説明を行ったが、当業者は本発明の趣旨、及び範囲内で形式、及び詳細を様々に変更できることが理解できよう。

【0093】

【発明の効果】 BEP におけるトランザクション階級の優先順位を動的に変更処理し、ほとんどの従来技術よりも低いオーバーヘッドを提供する。

【図面の簡単な説明】

【図1】本発明の好ましい実施例であるシステムのブロック図である。

【図2】トランザクション・サーバにおいて、マルチプログラミング・レベルに入るために、またはCPU処理

をするために待機するトランザクションの様々な待ち行列を示すブロック図である。

【図3】好ましい実施例に従って現平均階級応答時間の計算を例示する図である。

【図4】好ましい実施例のトランザクション・サーバによって使用されるスケジューリング・アルゴリズムの流れ図である。

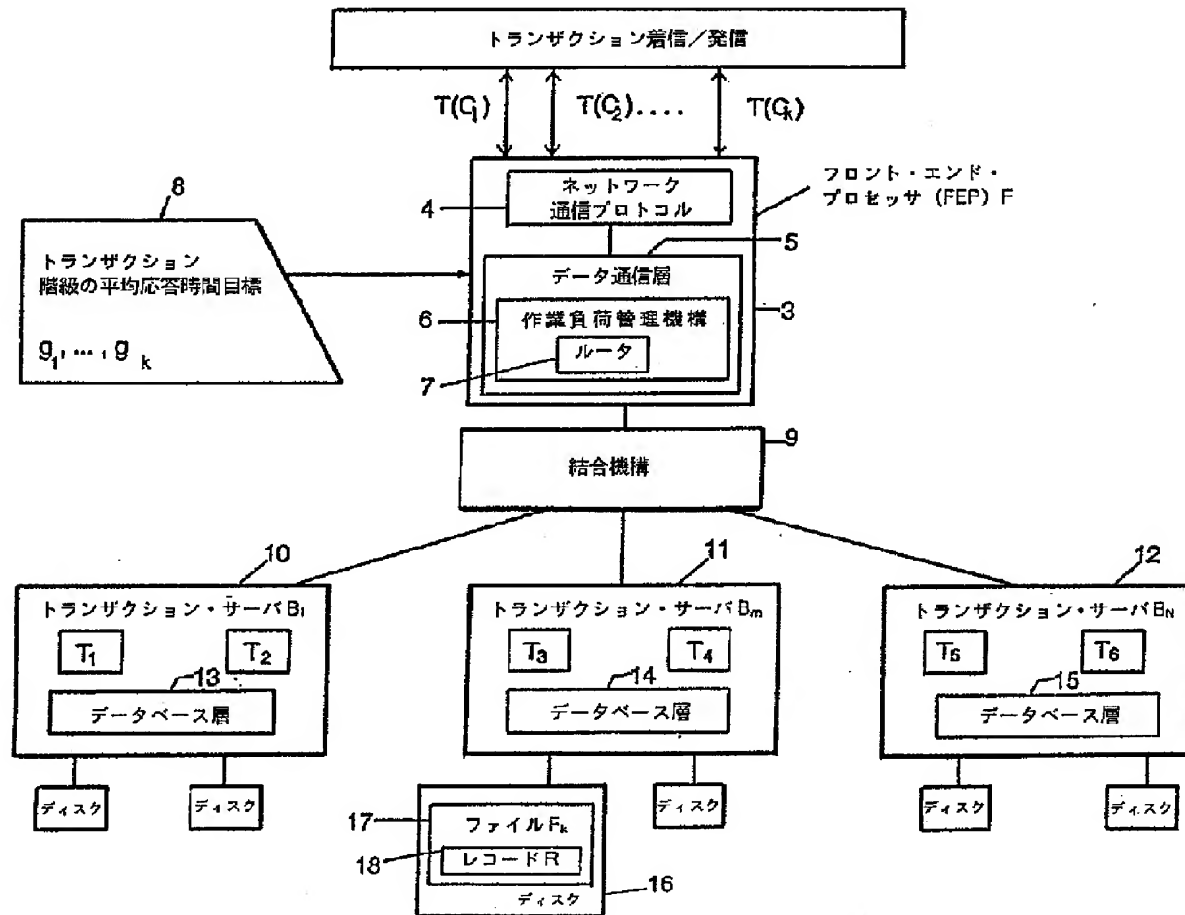
【図5】着信トランザクションに対する新しい階級性能指標の好ましい実施例による予測を例示する図である。

【図6】トランザクションの処理完了時における、現平均階級応答時間の好ましい実施例の更新を例示する図である。

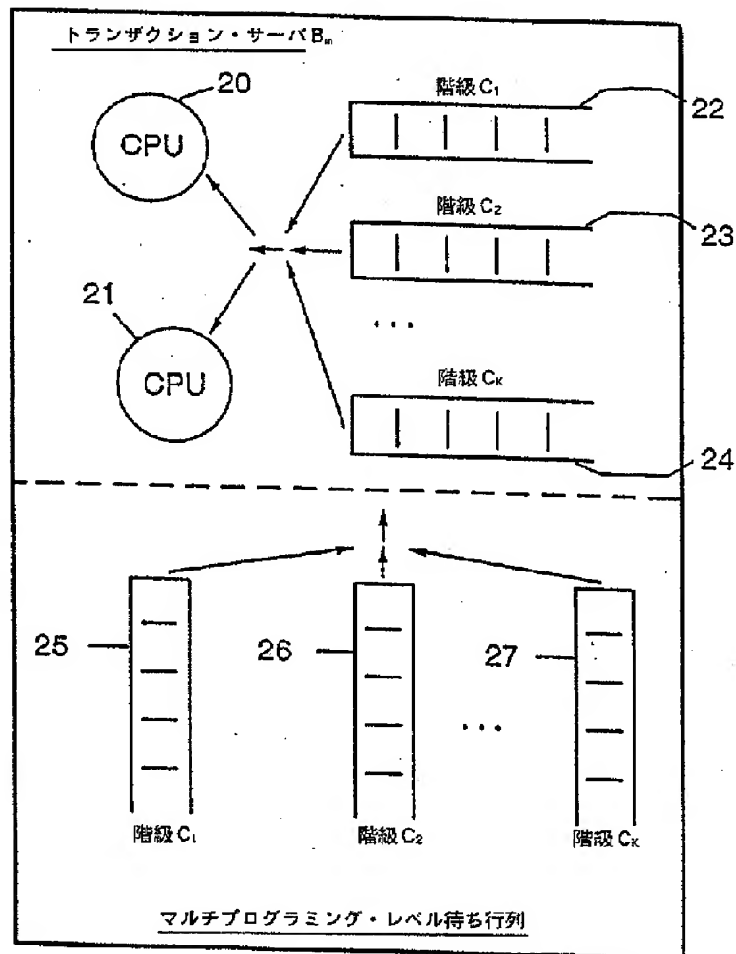
【符号の説明】

- 3 フロント・エンド・プロセッサ (FEP) F
- 4 ネットワーク通信プロトコル
- 5 データ通信層
- 6 作業負荷管理プログラム
- 7 ルータ
- 9 結合機構
- 10 トランザクション・サーバB_i
- 11 トランザクション・サーバB_m
- 12 トランザクション・サーバB_N
- 17 ファイルF_k
- 18 レコードR
- 22 階級C₁

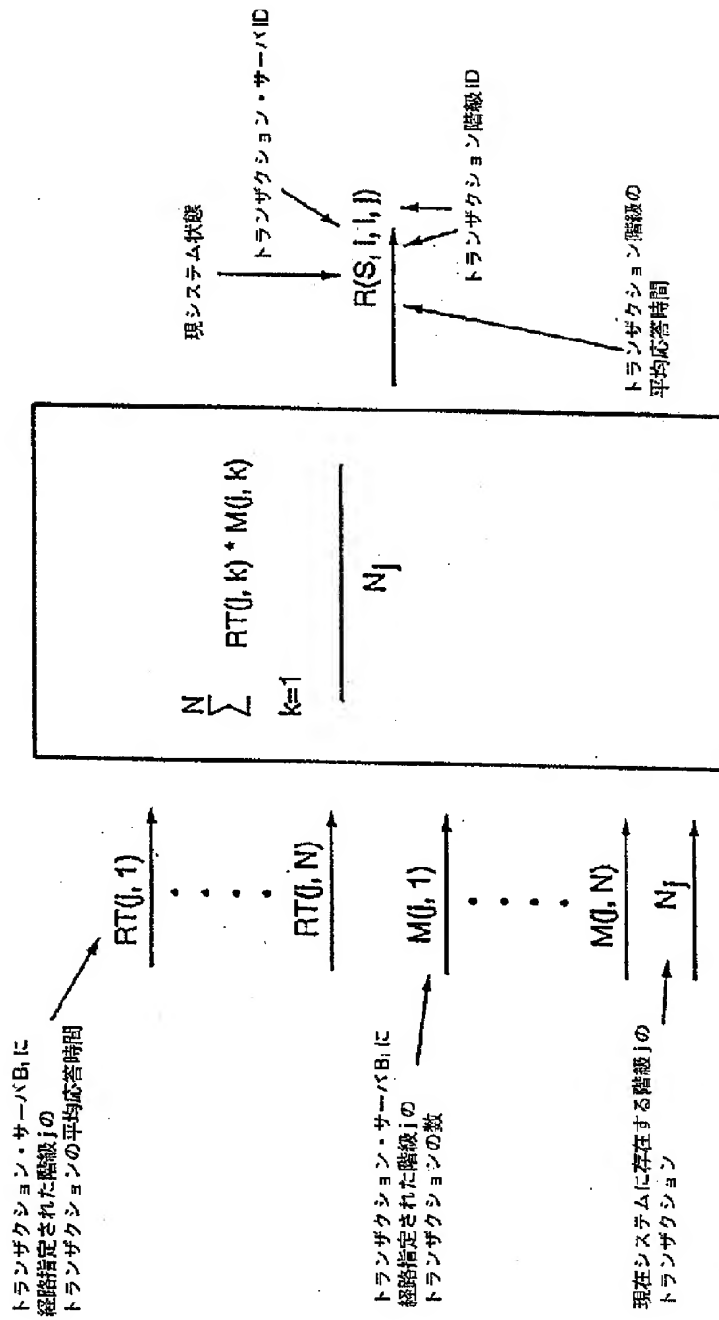
【図1】



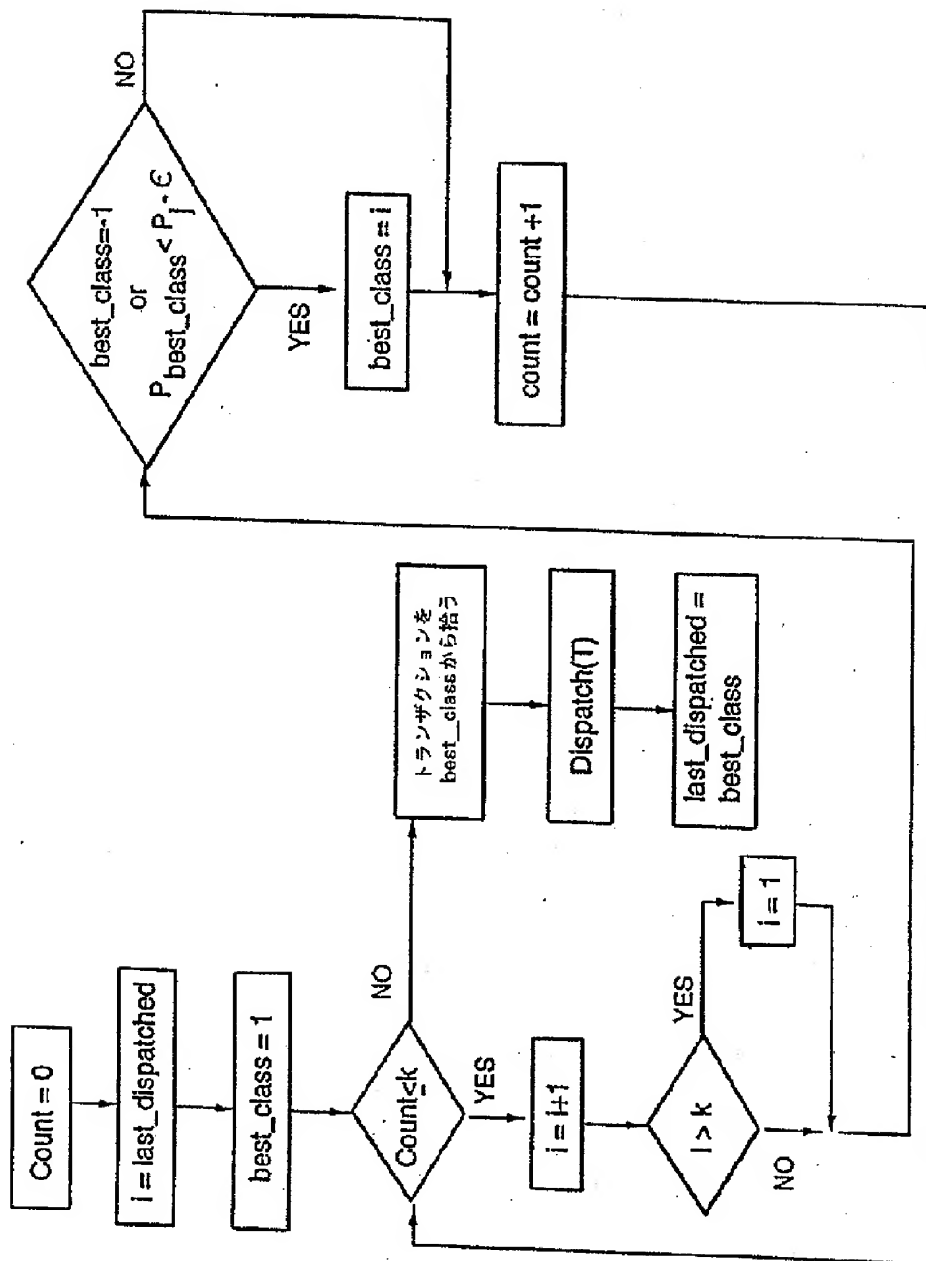
【図2】



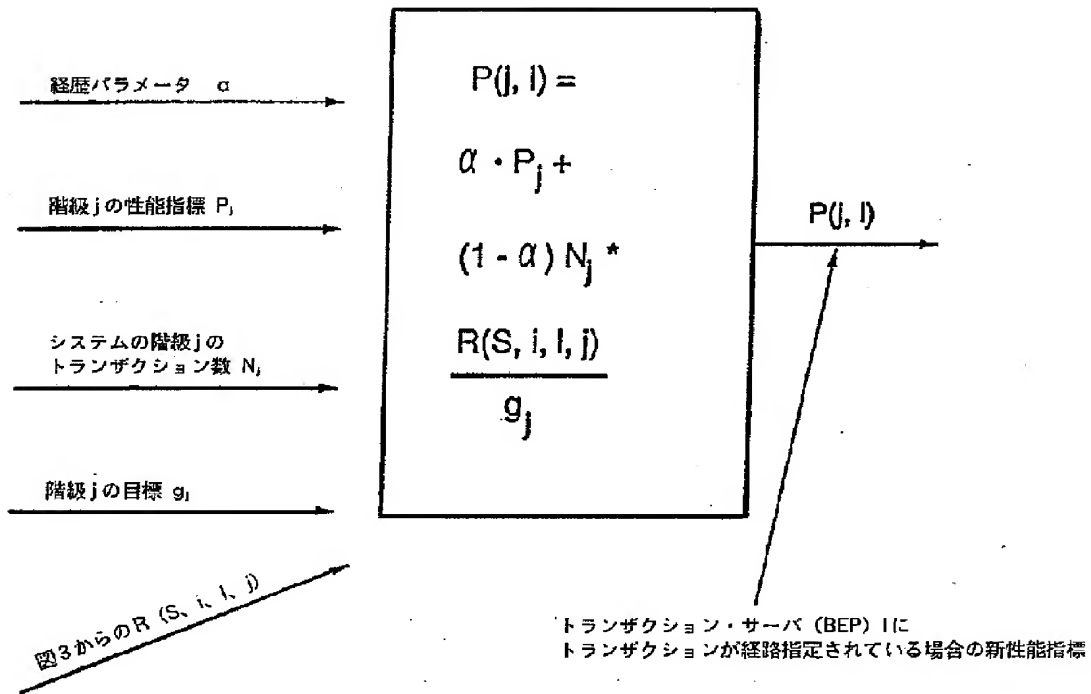
【図3】



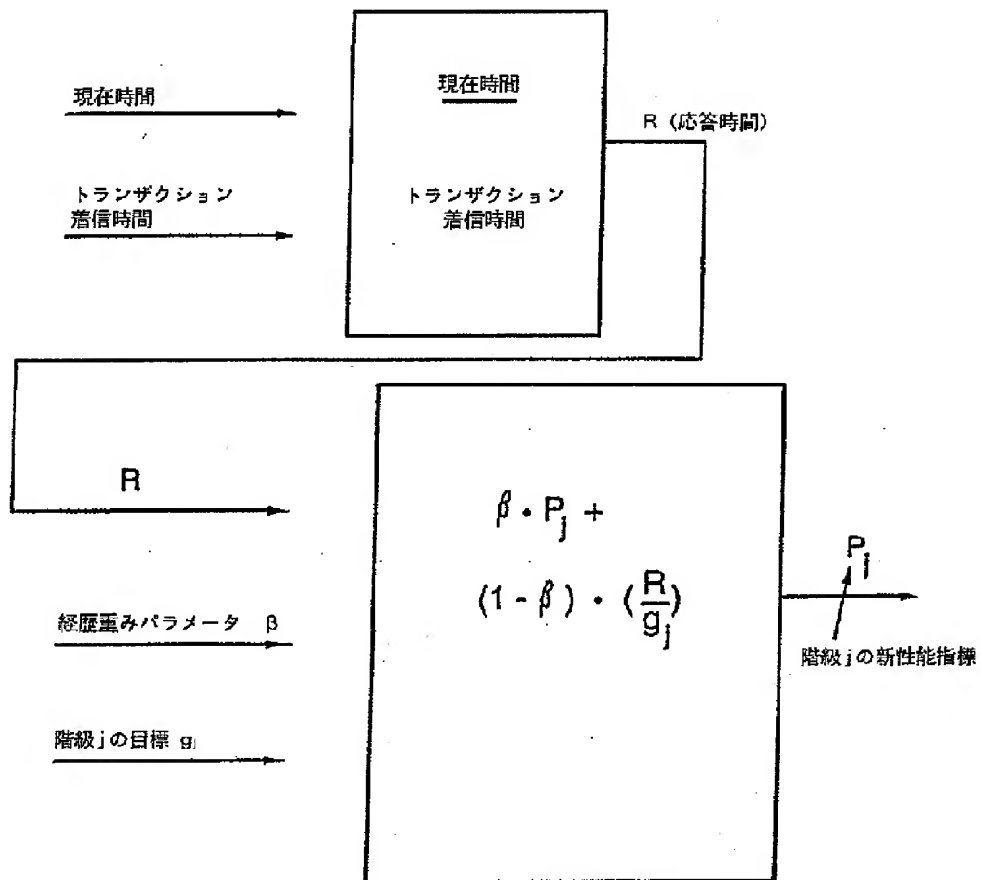
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 レオニダス・ジョージアディス
アメリカ合衆国10514、ニューヨーク州チ
ャパクア、オールド・ミル・ロード 38

(72)発明者 クリストス・ニコラス・ニコロー
アメリカ合衆国10023、ニューヨーク州ニ
ューヨーク、アパートメント 12イー、リ
パーサイド・ドライブ 5